

# 智聆口语评测

## API 文档

## 产品文档



腾讯云

---

**【版权声明】**

©2013-2019 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

**【商标声明】**

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

**【服务声明】**

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

## 文档目录

### API 文档

更新历史

简介

API 概览

调用方式

请求结构

公共参数

接口鉴权 v3

接口鉴权

返回结果

功能接口

发音评估初始化

发音数据传输接口

发音数据传输接口附带初始化过程

数据结构

错误码

# API 文档

## 更新历史

最近更新时间：2019-01-18 15:58:33

### 第 7 次发布

发布时间：2019-01-18 15:57:49

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [TransmitOralProcessWithInit](#)

修改接口：

- [InitOralProcess](#)
  - 新增入参：IsAsync
- [TransmitOralProcess](#)
  - 新增入参：IsQuery
  - 新增出参：Status

### 第 6 次发布

发布时间：2018-12-13 16:02:00

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [InitOralProcess](#)
  - 新增入参：SentenceInfoEnabled, ServerType
- [TransmitOralProcess](#)
  - 新增出参：SentenceInfoSet

新增数据结构：

- [SentenceInfo](#)

### 第 5 次发布

发布时间：2018-11-29 20:16:45

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [InitOralProcess](#)
  - 新增入参：StorageMode
- [TransmitOralProcess](#)
  - 新增出参：AudioUrl

## 第 4 次发布

发布时间：2018-10-29 11:21:16

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [InitOralProcess](#)
  - 新增入参：SoeAppld, IsLongLifeSession
  - 新增出参：SessionId
- [TransmitOralProcess](#)
  - 新增入参：IsLongLifeSession
  - 新增出参：SessionId

## 第 3 次发布

发布时间：2018-09-20 19:44:06

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [TransmitOralProcess](#)
  - 新增入参：SoeAppld

## 第 2 次发布

发布时间：2018-09-07 11:09:02

本次发布包含了以下内容：

改善已有的文档。

## 第 1 次发布

发布时间：2018-08-28 20:08:16

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [InitOralProcess](#)
- [TransmitOralProcess](#)

新增数据结构：

- [PhoneInfo](#)
- [WordRsp](#)

## 简介

最近更新时间：2019-01-18 15:58:30

**智聆口语评测 API 升级到 3.0 版本，全新的 API 接口文档更加规范和全面，统一的参数风格和公共错误码，统一的 SDK/CLI 版本与 API 文档严格一致，给您带来简单快捷的使用体验；支持全地域就近接入让您更快连接腾讯云产品。**

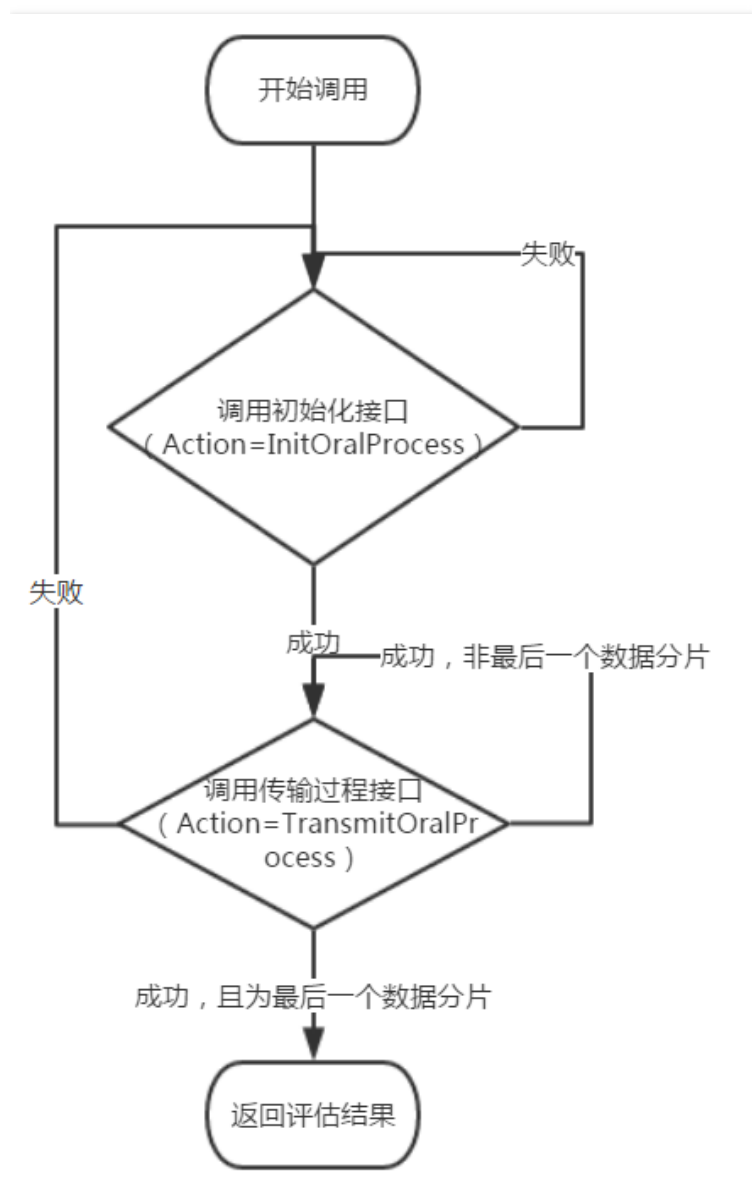
腾讯云智聆口语评测（Smart Oral Evaluation）英语口语评测服务，是基于英语口语类教育培训场景和腾讯云的语音处理技术，应用特征提取、声学模型和语音识别算法，为儿童和成人提供高准确度的英语口语发音评测。支持单词和句子模式的评测，多维度反馈口语表现。支持单词和句子评测模式，可广泛应用于英语口语类教学应用中。

## API 快速入门

本接口API实现对一段语音A和一段文本B进行发音评估，评估将该任务分为两个过程进行：

- 1）初始化过程：将文本B和工作模型（流式或非流式）通过API告知服务；
- 2）评估过程：以流式（将语音A分成若干段请求提供给服务）或非流式（将语音A一次性请求给服务）的方式请求服务，并在最后一次POST请求后收取评估结果（将isEnd设置为1）。

调用参数详情请见API功能接口文档，调用流程如下图：





## API 概览

最近更新时间：2019-01-18 15:58:31

### 功能接口

接口名称	接口功能
<a href="#">InitOralProcess</a>	发音评估初始化
<a href="#">TransmitOralProcess</a>	发音数据传输接口
<a href="#">TransmitOralProcessWithInit</a>	发音数据传输接口附带初始化过程

# 调用方式

## 请求结构

最近更新时间：2019-02-26 17:02:17

### 1. 服务地址

API 支持就近地域接入，本产品就近地域接入域名为 `soe.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `soe.ap-guangzhou.tencentcloudapi.com`。

推荐使用就近地域接入域名。根据调用接口时客户端所在位置，会自动解析到最近的某个具体地域的服务器。例如在广州发起请求，会自动解析到广州的服务器，效果和指定 `soe.ap-guangzhou.tencentcloudapi.com` 是一致的。

**注意：对时延敏感的业务，建议指定带地域的域名。**

目前支持的域名列表为：

接入地域	域名
就近地域接入（推荐，只支持非金融区）	<code>soe.tencentcloudapi.com</code>
华南地区(广州)	<code>soe.ap-guangzhou.tencentcloudapi.com</code>
华东地区(上海)	<code>soe.ap-shanghai.tencentcloudapi.com</code>
华北地区(北京)	<code>soe.ap-beijing.tencentcloudapi.com</code>
西南地区(成都)	<code>soe.ap-chengdu.tencentcloudapi.com</code>
西南地区(重庆)	<code>soe.ap-chongqing.tencentcloudapi.com</code>
东南亚地区(中国香港)	<code>soe.ap-hongkong.tencentcloudapi.com</code>
东南亚地区(新加坡)	<code>soe.ap-singapore.tencentcloudapi.com</code>
亚太地区(曼谷)	<code>soe.ap-bangkok.tencentcloudapi.com</code>
亚太地区(孟买)	<code>soe.ap-mumbai.tencentcloudapi.com</code>
亚太地区(首尔)	<code>soe.ap-seoul.tencentcloudapi.com</code>
亚太地区(东京)	<code>soe.ap-tokyo.tencentcloudapi.com</code>
美国东部(弗吉尼亚)	<code>soe.na-ashburn.tencentcloudapi.com</code>
美国西部(硅谷)	<code>soe.na-siliconvalley.tencentcloudapi.com</code>
北美地区(多伦多)	<code>soe.na-toronto.tencentcloudapi.com</code>
欧洲地区(法兰克福)	<code>soe.eu-frankfurt.tencentcloudapi.com</code>
欧洲地区(莫斯科)	<code>soe.eu-moscow.tencentcloudapi.com</code>

注意：由于金融区和非金融区是隔离不互通的，因此当访问金融区服务时（公共参数 Region 为金融区地域），需要同时指定带金融区地域的域名，最好和 Region 的地域保持一致。

金融区接入地域	金融区域名
华东地区(上海金融)	soe.ap-shanghai-fsi.tencentcloudapi.com
华南地区(深圳金融)	soe.ap-shenzhen-fsi.tencentcloudapi.com

## 2. 通信协议

腾讯云 API 的所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道。

## 3. 请求方法

支持的 HTTP 请求方法:

- POST ( 推荐 )
- GET

POST 请求支持的 Content-Type 类型：

- application/json ( 推荐 )，必须使用 TC3-HMAC-SHA256 签名方法。
- application/x-www-form-urlencoded，必须使用 HmacSHA1 或 HmacSHA256 签名方法。
- multipart/form-data ( 仅部分接口支持 )，必须使用 TC3-HMAC-SHA256 签名方法。

GET 请求长度不得超过 32 KB。POST 请求使用签名方法为 HmacSHA1、HmacSHA256 时长度不得超过 1 MB。

## 4. 字符编码

均使用UTF-8编码。

## 公共参数

最近更新时间：2019-02-26 17:02:17

公共参数是用于标识用户和接口鉴权目的的参数，如非必要，在每个接口单独的接口文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

## 签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	具体操作的指令接口名称，例如想要调用云服务器的查询实例列表接口，则 Action 参数即为 DescribeInstances。
X-TC-Region	String	是	地域参数，用来标识希望操作哪个地域的数据。
X-TC-Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前API服务器时间相差超过5分钟，会引起签名过期错误。
X-TC-Version	String	是	API 的版本。例如 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部； - Signature：签名摘要。
X-TC-Token	String	否	临时证书所用的 Token，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

假设用户想要查询广州地域的云服务器实例列表，则其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Limit=10&Offset=0
```

```
Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WfkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Host: cvm.tencentcloudapi.com
```

```
X-TC-Action: DescribeInstances
```

```
X-TC-Version: 2017-03-12
```

```
X-TC-Timestamp: 1539084154
X-TC-Region: ap-guangzhou
```

HTTP POST ( application/json ) 请求结构示例：

```
https://cvm.tencentcloudapi.com/

Authorization: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: application/json
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

{"Offset":0,"Limit":10}
```

HTTP POST ( multipart/form-data ) 请求结构示例 ( 仅特定的接口支持 )：

```
https://cvm.tencentcloudapi.com/

Authorization: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: multipart/form-data; boundary=58731222010402
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

--58731222010402
Content-Disposition: form-data; name="Offset"

0
--58731222010402
Content-Disposition: form-data; name="Limit"

10
--58731222010402--
```

## 签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	具体操作的指令接口名称，例如想要调用云服务器的查询实例列表接口，则 Action 参数即为 DescribeInstances。
Region	String	是	地域参数，用来标识希望操作哪个地域的数据。

参数名称	类型	必选	描述
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在 <a href="#">云API密钥</a> 上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	API 的版本。例如 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

假设用户想要查询广州地域的云服务器实例列表，其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKIDEXAMPLE
```

```
Host: cvm.tencentcloudapi.com
Content-Type: application/x-www-form-urlencoded
```

HTTP POST 请求结构示例：

```
https://cvm.tencentcloudapi.com/
```

```
Host: cvm.tencentcloudapi.com
Content-Type: application/x-www-form-urlencoded
```

```
Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKIDEXAMPLE
```

# 接口鉴权 v3

最近更新时间：2019-02-26 17:02:18

腾讯云 API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往[云API密钥页面](#)申请，否则无法调用云API接口。

## 申请安全凭证

在第一次使用云API之前，请前往[云API密钥页面](#)申请安全凭证。安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- **用户必须严格保管安全凭证，避免泄露。**

申请安全凭证的具体步骤如下：

1. 登录[腾讯云管理中心控制台](#)。
2. 前往[云API密钥](#)的控制台页面
3. 在[云API密钥](#)页面，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

## TC3-HMAC-SHA256 签名方法

注意：对于GET方法，只支持 Content-Type: application/x-www-form-urlencoded 协议格式。对于POST方法，目前支持 Content-Type: application/json 以及 Content-Type: multipart/form-data 两种协议格式，json 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，此时该接口不能使用 json 格式调用，参考具体业务接口文档说明。

下面以云服务器查询广州实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

### 1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

```
CanonicalRequest =  
HTTPRequestMethod + '\n' +  
CanonicalURI + '\n' +  
CanonicalQueryString + '\n' +  
CanonicalHeaders + '\n' +  
SignedHeaders + '\n' +  
HashedRequestPayload
```

- HTTPRequestMethod：HTTP 请求方法（GET、POST），本示例中为 GET；

- CanonicalURI：URI 参数，API 3.0 固定为正斜杠 (/)；
- CanonicalQueryString：发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串，对于 GET 请求，则为 URL 中间号 (?) 后面的字符串内容，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1) 头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2) 多个头部，按照头部 key (小写) 的字典排序进行拼接。此例中为：content-type:application/x-www-form-urlencoded\nhost:cvm.tencentcloudapi.com\n
- SignedHeaders：参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。content-type 和 host 为必选头部。拼接规则：1) 头部 key 统一转成小写；2) 多个头部 key (小写) 按照字典排序进行拼接，并且以分号 (;) 分隔。此例中为：content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.api.tencentyun.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

## 2. 拼接待签名字符串

按如下格式拼接待签名字符串：

```
StringToSign =
Algorithm + \n +
RequestTimestamp + \n +
CredentialScope + \n +
HashedCanonicalRequest
```

- Algorithm：签名算法，目前固定为 TC3-HMAC-SHA256；
- RequestTimestamp：请求时间戳，即请求头部的 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- CredentialScope：凭证范围，格式为 Date/service/tc3\_request，包含日期、所请求的服务和终止字符串 (tc3\_request)。Date 为 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为产品名，必须与调用的产品域名一致，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3\_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

1. Date 必须从时间戳 X-TC-Timestamp 计算得到，且时区为 UTC+0。如果加入系统本地时区信息，例如东八区，将导致白天和晚上调用成功，但是凌晨调用必定失败。假设时间戳为 1551113065，在东八区的时间是 2019-02-26 00:44:25，但是计算得



到的 Date 取 UTC+0 的日期应为 2019-02-25，而不是 2019-02-26。

2. Timestamp 必须是当前系统时间，且需确保系统时间和标准时间是同步的，如果相差超过五分钟则必定失败。如果长时间不和标准时间同步，可能导致运行一段时间后，请求必定失败（返回签名过期错误）。

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

### 3. 计算签名

1) 计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为 2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2) 计算签名，伪代码如下

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

### 4. 拼接 Authorization

按如下格式拼接 Authorization：

```
Authorization =
Algorithm + ' ' +
'Credential=' + SecretId + '/' + CredentialScope + ', ' +
'SignedHeaders=' + SignedHeaders + ', '
'Signature=' + Signature
```

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3\_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474

最终完整的调用信息如下：

`https://cvm.tencentcloudapi.com/?Limit=10&Offset=0`

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3\_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474

Content-Type: application/x-www-form-urlencoded

Host: cvm.tencentcloudapi.com

X-TC-Action: DescribeInstances

X-TC-Version: 2017-03-12

X-TC-Timestamp: 1539084154

X-TC-Region: ap-guangzhou

## 5. 签名演示

### Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class TencentCloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.tencentcloudapi.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URL_ENCODED = "application/x-www-form-urlencoded";
    private final static String CT_JSON = "application/json";
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(byte[] key, String msg) throws Exception {
        Mac mac = Mac.getInstance("HmacSHA256");
        SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
        mac.init(secretKeySpec);
        return mac.doFinal(msg.getBytes(CHARSET));
    }
}
```

```
}

public static void main(String[] args) throws Exception {
    String service = "cvm";
    String host = "cvm.tencentcloudapi.com";
    String region = "ap-guangzhou";
    String action = "DescribeInstances";
    String version = "2017-03-12";
    String algorithm = "TC3-HMAC-SHA256";
    String timestamp = "1539084154";
    //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    // 注意时区，否则容易出错
    sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
    String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

    // ***** 步骤 1：拼接规范请求串 *****
    String httpRequestMethod = "GET";
    String canonicalUri = "/";
    String canonicalQueryString = "Limit=10&Offset=0";
    String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host + "\n";
    String signedHeaders = "content-type;host";
    String hashedRequestPayload = DigestUtils.sha256Hex("");
    String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryString + "\n"
        + canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
    System.out.println(canonicalRequest);

    // ***** 步骤 2：拼接待签名字符串 *****
    String credentialScope = date + "/" + service + "/" + "tc3_request";
    String hashedCanonicalRequest = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
    String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
    System.out.println(stringToSign);

    // ***** 步骤 3：计算签名 *****
    byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
    byte[] secretService = sign256(secretDate, service);
    byte[] secretSigning = sign256(secretService, "tc3_request");
    String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLowerCase();
    System.out.println(signature);

    // ***** 步骤 4：拼接 Authorization *****
    String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
        + "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
    System.out.println(authorization);

    TreeMap<String, String> headers = new TreeMap<String, String>();
    headers.put("Authorization", authorization);
    headers.put("Host", host);
    headers.put("Content-Type", CT_X_WWW_FORM_URL_ENCODED);
    headers.put("X-TC-Action", action);
    headers.put("X-TC-Timestamp", timestamp);
    headers.put("X-TC-Version", version);
    headers.put("X-TC-Region", region);
}
```

```
}  
}
```

## Python

```
# -*- coding: utf-8 -*-  
import hashlib, hmac, json, os, sys, time  
from datetime import datetime  
  
# 密钥参数  
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"  
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"  
  
service = "cvm"  
host = "cvm.tencentcloudapi.com"  
endpoint = "https://" + host  
region = "ap-guangzhou"  
action = "DescribeInstances"  
version = "2017-03-12"  
algorithm = "TC3-HMAC-SHA256"  
timestamp = 1539084154  
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")  
params = {"Limit": 10, "Offset": 0}  
  
# ***** 步骤 1：拼接规范请求串 *****  
http_request_method = "GET"  
canonical_uri = "/"  
canonical_querystring = "Limit=10&Offset=0"  
ct = "x-www-form-urlencoded"  
payload = ""  
if http_request_method == "POST":  
    canonical_querystring = ""  
    ct = "json"  
    payload = json.dumps(params)  
canonical_headers = "content-type:application/%s\nhost:%s\n" % (ct, host)  
signed_headers = "content-type;host"  
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()  
canonical_request = (http_request_method + "\n" +  
    canonical_uri + "\n" +  
    canonical_querystring + "\n" +  
    canonical_headers + "\n" +  
    signed_headers + "\n" +  
    hashed_request_payload)  
print(canonical_request)  
  
# ***** 步骤 2：拼接待签名字符串 *****  
credential_scope = date + "/" + service + "/" + "tc3_request"  
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()  
string_to_sign = (algorithm + "\n" +  
    str(timestamp) + "\n" +  
    credential_scope + "\n" +  
    hashed_canonical_request)  
print(string_to_sign)
```

```
# ***** 步骤 3 : 计算签名 *****
# 计算签名摘要函数
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ***** 步骤 4 : 拼接 Authorization *****
authorization = (algorithm + " " +
"Credential=" + secret_id + "/" + credential_scope + ", " +
"SignedHeaders=" + signed_headers + ", " +
"Signature=" + signature)
print(authorization)

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

## 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

# 接口鉴权

最近更新时间：2019-02-26 17:02:18

腾讯云 API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往[云API密钥页面](#)申请，否则无法调用云API接口。

## 1. 申请安全凭证

在第一次使用云API之前，请前往[云API密钥页面](#)申请安全凭证。安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- **用户必须严格保管安全凭证，避免泄露。**

申请安全凭证的具体步骤如下：

1. 登录[腾讯云管理中心控制台](#)。
2. 前往[云API密钥](#)的控制台页面
3. 在[云API密钥](#)页面，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

## 2. 生成签名串

有了安全凭证SecretId 和 SecretKey后，就可以生成签名串了。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

**注意：这里只是示例，请根据用户实际申请的 SecretId 和 SecretKey 进行后续操作！**

以云服务器查看实例列表(DescribeInstances)请求为例，当用户调用这一接口时，其请求参数可能如下：

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	ap-guangzhou
InstanceId.0	待查询的实例ID	ins-09dx96dg

参数名称	中文	参数值
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

## 2.1. 对参数排序

首先对所有请求参数按参数名的字典序（ASCII 码）升序排序。注意：1）只按参数名进行排序，参数值保持对应即可，不参与比大小；2）按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。用户可以借助编程语言中的相关排序函数来实现这一功能，如 php 中的 ksort 函数。上述示例参数的排序结果如下：

```
{
  'Action': 'DescribeInstances',
  'InstanceIds.0': 'ins-09dx96dg',
  'Limit': 20,
  'Nonce': 11886,
  'Offset': 0,
  'Region': 'ap-guangzhou',
  'SecretId': 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
  'Timestamp': 1465185768,
  'Version': '2017-03-12',
}
```

使用其它程序设计语言开发时，可对上面示例中的参数进行排序，得到的结果一致即可。

## 2.2. 拼接请求字符串

此步骤生成请求字符串。将把上一步排序好的请求参数格式化成“参数名称”=“参数值”的形式，如对 Action 参数，其参数名称为 "Action"，参数值为 "DescribeInstances"，因此格式化后就为 Action=DescribeInstances。注意：“参数值”为原始值而非url编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

## 2.3. 拼接签名原字符串

此步骤生成签名原字符串。签名原字符串由以下几个参数构成：

1. 请求方法: 支持 POST 和 GET 方式，这里使用 GET 请求，注意方法为全大写。
2. 请求主机: 查看实例列表(DescribeInstances)的请求域名为：cvm.tencentcloudapi.com。实际的请求域名根据接口所属模块的不同而不同，详见各接口说明。
3. 请求路径: 当前版本云API的请求路径固定为 /。
4. 请求字符串: 即上一步生成的请求字符串。

签名原文串的拼接规则为: 请求方法 + 请求主机 + 请求路径 + ? + 请求字符串

示例的拼接结果为：

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

## 2.4. 生成签名串

此步骤生成签名串。首先使用 HMAC-SHA1 算法对上一步中获得的**签名原文字符串**进行签名，然后将生成的签名串使用 Base64 进行编码，即可获得最终的签名串。

具体代码如下，以 PHP 语言为例：

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为：

```
EliP9YW3pW28FpsEdkXt/+WcGel=
```

使用其它程序设计语言开发时，可用上面示例中的原文进行签名验证，得到的签名串与例子中的一致即可。

## 3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

如上一步生成的签名串为 EliP9YW3pW28FpsEdkXt/+WcGel=，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt/+WcGel=，它将用于生成最终的请求 URL。

**注意：**如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则发送请求时所有请求参数的值均需要做 URL 编码，参数键和=符号不需要编码。非 ASCII 字符在 URL 编码前需要先用 UTF-8 进行编码。

**注意：**有些编程语言的 http 库会自动为所有参数进行 urlencode，在这种情况下，就不需要对签名串进行 URL 编码了，否则两次 URL 编码会导致签名失败。

**注意：**其他参数值也需要进行编码，编码采用 RFC 3986。使用 %XY 对特殊字符例如汉字进行百分比编码，其中“X”和“Y”为十六进制字符（0-9 和大写字母 A-F），使用小写将引发错误。

## 4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在



错误代码	错误描述
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

## 5. 签名演示

在实际调用 API 3.0 时，推荐使用配套的腾讯云 SDK 3.0，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 [SDK 中心](#)。当前支持的编程语言有：

- [Python](#)
- [Java](#)
- [PHP](#)
- [Go](#)
- [JavaScript](#)
- [.NET](#)

为了更清楚的解释签名过程，下面以实际编程语言为例，将上述的签名过程具体实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK。

最终输出的 url 可能为：`https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=Elip9YW3pW28FpsEdkXt/+WcGel=&Timestamp=1465185768&Version=2017-03-12`

注意：由于示例中的密钥是虚构的，时间戳也不是系统当前时间，因此如果将此 url 在浏览器中打开或者用 curl 等命令调用时会返回鉴权错误：签名过期。为了得到一个可以正常返回的 url，需要修改示例中的 SecretId 和 SecretKey 为真实的密钥，并使用系统当前时间戳作为 Timestamp。

注意：在下面的示例中，不同编程语言，甚至同一语言每次执行得到的 url 可能都有所不同，表现为参数的顺序不同，但这并不影响正确性。只要所有参数都在，且签名计算正确即可。

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，即使是旧版的 API，由于存在细节差异也会导致签名计算错误，请以对应的实际文档为准。

### Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPIDemo {
    private final static String CHARSET = "UTF-8";
```

```
public static String sign(String s, String key, String method) throws Exception {
    Mac mac = Mac.getInstance(method);
    SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
    mac.init(secretKeySpec);
    byte[] hash = mac.doFinal(s.getBytes(CHARSET));
    return DatatypeConverter.printBase64Binary(hash);
}

public static String getStringToSign(TreeMap<String, Object> params) {
    StringBuilder s2s = new StringBuilder("GETcvm.tencentcloudapi.com/?");
    // 签名时要求对参数进行字典排序, 此处用TreeMap保证顺序
    for (String k : params.keySet()) {
        s2s.append(k).append("=").append(params.get(k).toString()).append("&");
    }
    return s2s.toString().substring(0, s2s.length() - 1);
}

public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException {
    StringBuilder url = new StringBuilder("https://cvm.tencentcloudapi.com/?");
    // 实际请求的url中对参数顺序没有要求
    for (String k : params.keySet()) {
        // 需要对请求串进行urlencode, 由于key都是英文字母, 故此处仅对其value进行urlencode
        url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
    }
    return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
    TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
    // 实际调用时应当使用随机数, 例如: params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
    params.put("Nonce", 11886); // 公共参数
    // 实际调用时应当使用系统当前时间, 例如: params.put("Timestamp", System.currentTimeMillis() / 1000);
    params.put("Timestamp", 1465185768); // 公共参数
    params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 公共参数
    params.put("Action", "DescribeInstances"); // 公共参数
    params.put("Version", "2017-03-12"); // 公共参数
    params.put("Region", "ap-guangzhou"); // 公共参数
    params.put("Limit", 20); // 业务参数
    params.put("Offset", 0); // 业务参数
    params.put("InstanceId.0", "ins-09dx96dg"); // 业务参数
    params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE", "HmacSHA1")); // 公共参数
    System.out.println(getUrl(params));
}
}
```

## Python

注意：如果是在 Python 2 环境中运行，需要先安装 requests 依赖包： `pip install requests`。

```
# -*- coding: utf8 -*-
import base64
```

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "/"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if __name__ == '__main__':
    endpoint = "cvm.tencentcloudapi.com"
    data = {
        'Action': 'DescribeInstances',
        'InstanceId.0': 'ins-09dx96dg',
        'Limit': 20,
        'Nonce': 11886,
        'Offset': 0,
        'Region': 'ap-guangzhou',
        'SecretId': secret_id,
        'Timestamp': 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # 此处会实际调用，成功后可能产生计费
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```

## 返回结果

最近更新时间：2019-02-26 17:02:19

### 正确返回结果

以云服务器的接口查看实例状态列表 (DescribeInstancesStatus) 2017-03-12 版本为例，若调用成功，其可能的返回如下为：

```
{
  "Response": {
    "TotalCount": 0,
    "InstanceStatusSet": [],
    "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
  }
}
```

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。
- 除了固定的字段外，其余均为具体接口定义的字段，不同的接口所返回的字段参见接口文档中的定义。此例中的 TotalCount 和 InstanceStatusSet 均为 DescribeInstancesStatus 接口定义的字段，由于调用请求的用户暂时还没有云服务器实例，因此 TotalCount 在此情况下的返回值为 0，InstanceStatusSet 列表为空。

### 错误返回结果

若调用失败，其返回值示例如下为：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

### 公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s)请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

# 功能接口

## 发音评估初始化

最近更新时间：2019-01-18 15:58:34

### 1. 接口描述

接口请求域名：soe.tencentcloudapi.com。

初始化发音评估过程，每一轮评估前进行调用。语音输入模式分为流式模式和非流式模式，流式模式支持数据分片传输，可以加快评估响应速度。评估模式分为词模式和句子模式，词模式会标注每个音节的详细信息；句子模式会有完整度和流利度的评估。

默认接口请求频率限制：10000次/秒。

### 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：InitOralProcess
Version	是	String	公共参数，本接口取值：2018-07-24
Region	否	String	公共参数，本接口不需要传递此参数。
SessionId	是	String	语音段唯一标识，一段语音一个SessionId
RefText	是	String	被评估语音对应的文本，句子模式下不超过个 20 单词或者中文文字，段落模式不超过 120 单词或者中文文字，中文评估使用 utf-8 编码，自由说模式该值传空。
WorkMode	是	Integer	语音输入模式，0：流式分片，1：非流式一次性评估
EvalMode	是	Integer	评估模式，0：词模式，1：句子模式，2：段落模式，3：自由说模式，当为词模式评估时，能够提供每个音节的评估信息，当为句子模式时，能够提供完整度和流利度信息。
ScoreCoeff	是	Float	评价苛刻指数，取值为[1.0 - 4.0]范围内的浮点数，用于平滑不同年龄段的分数，1.0 为小年龄段，4.0为最高年龄段
SoeAppId	否	String	业务应用ID，与账号应用APPID无关，是用来方便客户管理服务的参数，新的 SoeAppId 可以在 <a href="#">控制台</a> 【应用管理】下新建。
IsLongLifeSession	否	Integer	长效session标识，当该参数为1时，session的持续时间为300s，但会一定程度上影响第一个数据包的返回速度，且TransmitOralProcess必须同时为1才可生效。
StorageMode	否	Integer	音频存储模式，0：不存储，1：存储到公共对象存储，输出结果为该会话最后一个分片TransmitOralProcess 返回结果 AudioUrl 字段。

参数名称	必选	类型	描述
SentenceInfoEnabled	否	Integer	输出断句中间结果标识，0：不输出，1：输出，通过设置该参数，可以在评估过程中的分片传输请求中，返回已经评估断句的中间结果，中间结果可用于客户端 UI 更新，输出结果为TransmitOralProcess请求返回结果 SentenceInfoSet 字段。
ServerType	否	Integer	评估语言，0：英文，1：中文。
IsAsync	否	Integer	异步模式标识，0：同步模式，1：异步模式。

### 3. 输出参数

参数名称	类型	描述
SessionId	String	语音段唯一标识，一个完整语音一个SessionId
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

### 4. 示例

#### 示例1 初始化发音评估过程

初始化发音评估过程

##### 输入示例

```
https://soe.tencentcloudapi.com/?Action=InitOralProcess
&SessionId=stress_test_956938
&WorkMode=0
&RefText=again
&EvalMode=0
&ScoreCoeff=3.5
&<公共请求参数>
```

##### 输出示例

```
{
  "Response": {
    "RequestId": "xxxxxxx"
  }
}
```

### 5. 开发者资源

#### API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

## SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

## 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
InternalServerError	内部错误
InternalServerError.BASE64DecodeFailed	BASE64解码错误
InternalServerError.IllegalServerResponse	服务器应答非法
InternalServerError.InitialParameterError	初始化参数错误
InternalServerError.InvalidSeqId	分片序号错误。
InternalServerError.InvalidWAVHeader	WAV头部格式非法或不在同一分片内。
InternalServerError.NoConversationFound	使用的会话没有找到或已经被释放
InternalServerError.ServerInternalError	服务器内部错误
InternalServerError.ServerOverload	服务器过载
InternalServerError.ServiceTimeout	服务超时
InternalServerError.ShardNoStartWithOne	分片序号错误应该从1开始
InternalServerError.StreamingvoicepkgTimeout	流式语音包超时
InternalServerError.VoiceMsgOversized	语音数据大于1MB
InternalServerError.WordLengthTooLong	文本单词超过限制
InvalidParameter	参数错误
InvalidParameter.AuthorizeError	服务未开通或已欠费。



错误码	描述
InvalidParameter.InitialParameterError	初始化参数错误。
InvalidParameter.VoiceMsgOversized	语音数据大于1MB。
InvalidParameterValue	参数取值错误
InvalidParameterValue.BASEDecodeFailed	BASE64解码错误。
InvalidParameterValue.InvalidSeqId	分片序号错误。
InvalidParameterValue.InvalidWAVHeader	WAV头部格式非法或不在同一分片内。
InvalidParameterValue.NoDocInList	表单中没有文件。
InvalidParameterValue.RefTxtTooLang	输入文本太长。
InvalidParameterValue.ShardNoStartWithOne	分片序号错误应该从1开始。
InvalidParameterValue.StreamingvoicepkgTimeout	流式语音包超时。
InvalidParameterValue.VadNotDetectedSpeak	没有检测到语音。
InvalidParameterValue.WordLengthTooLong	文本单词超过限制。
ResourceUnavailable.AuthorizeError	服务未开通或已欠费。
ResourceUnavailable.CannotFindSession	评估之前没有初始化或已过期。
ResourceUnavailable.InitStreamNotSupport	该接口不支持init_stream。
ResourceUnavailable.InitStreamUnfinished	初始化请求未完成，请稍后重试。
ResourceUnavailable.LastSeqUnfinished	前一个分片未处理完，请稍后重试。
ResourceUnavailable.NoConversationFound	使用的会话没有找到或已经被释放。
ResourceUnavailable.NoInitBeforeEvaluation	评估之前没有初始化。

# 发音数据传输接口

最近更新时间：2019-01-18 15:58:33

## 1. 接口描述

接口请求域名：soe.tencentcloudapi.com。

传输音频数据，必须在完成发音评估初始化接口之后调用，且SessionId要与初始化接口保持一致。分片传输时，尽量保证SeqId顺序传输。音频源目前仅支持16k采样率16bit单声道编码方式，如有不一致可能导致评估不准确或失败。

默认接口请求频率限制：10000次/秒。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：TransmitOralProcess
Version	是	String	公共参数，本接口取值：2018-07-24
Region	否	String	公共参数，本接口不需要传递此参数。
SeqId	是	Integer	流式数据包的序号，从1开始，当IsEnd字段为1后后续序号无意义，当IsLongLifeSession不为1且为非流式模式时无意义。
IsEnd	是	Integer	是否传输完毕标志，若为0表示未完毕，若为1则传输完毕开始评估，非流式模式下无意义。
VoiceFileType	是	Integer	语音文件类型 1:raw, 2:wav, 3:mp3(三种格式目前仅支持16k采样率16bit编码单声道，如有不一致可能导致评估不准确或失败)。
VoiceEncodeType	是	Integer	语音编码类型 1:pcm。
UserVoiceData	是	String	当前数据包数据, 流式模式下数据包大小可以按需设置，数据包大小必须 $\geq 4K$ ，且必须保证分片帧完整（16bit的数据必须保证音频长度为偶数），编码格式要求为BASE64。
SessionId	是	String	语音段唯一标识，一个完整语音一个SessionId。
SoeAppId	否	String	业务应用ID，与账号应用APPID无关，是用来方便客户管理服务的参数，新的SoeAppId 可以在 <a href="#">控制台</a> 【应用管理】下新建。
IsLongLifeSession	否	Integer	长效session标识，当该参数为1时，session的持续时间为300s，但会一定程度上影响第一个数据包的返回速度。当InitOralProcess接口调用时此项为1时，此项必填1才可生效。
IsQuery	否	Integer	查询标识，当该参数为1时，该请求为查询请求，请求返回该 Session 的评估结果。

### 3. 输出参数

参数名称	类型	描述
PronAccuracy	Float	发音精准度，取值范围[-1, 100]，当取-1时指完全不匹配，当为句子模式时，是所有已识别单词准确度的加权平均值。当为流式模式且请求中IsEnd未置1时，取值无意义
PronFluency	Float	发音流利度，取值范围[0, 1]，当为词模式时，取值无意义；当为流式模式且请求中IsEnd未置1时，取值无意义
PronCompletion	Float	发音完整度，取值范围[0, 1]，当为词模式时，取值无意义；当为流式模式且请求中IsEnd未置1时，取值无意义
Words	Array of WordRsp	详细发音评估结果
SessionId	String	语音段唯一标识，一段语音一个SessionId
AudioUrl	String	保存语音音频文件下载地址
SentenceInfoSet	Array of SentenceInfo	断句中间结果，中间结果是局部最优而非全局最优的结果，所以中间结果有可能和最终整体结果对应部分不一致；中间结果的输出便于客户端UI更新；待用户发音完全结束后，系统会给出一个综合所有句子的整体结果。
Status	String	评估 session 状态，“Evaluating”：评估中、“Failed”：评估失败、“Finished”：评估完成
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

### 4. 示例

#### 示例1 传输语音进行发音评估

传输语音进行发音评估

##### 输入示例

```
https://soe.tencentcloudapi.com/?Action=TransmitOralProcess
&SessionId=stress_test_956938
&VoiceFileType=1
&SeqId=0
&VoiceEncodeType=1
&IsEnd=0
&UserVoiceData=VWtsR1JxeUpBd0JYUVZaRlptMTBJQkFBQUFBQkFBRUFnRDRBQUFCOUFBQUUNBQkFBVEVsVFZCb0FBQUJ
KVGtaUFNWTKdWQTRBQUFCTVIYWm1OVFI1TVRrdU1UQXkdBR1JoZEdGbWIRTUF5
&<公共请求参数>
```

##### 输出示例

```
{
  "Response": {
    "PronAccuracy": 0.99,
    "PronFluency": 0.99,
```

```
"PronCompletion": 0.99,
"RequestId": "xxxxxxx",
"Words": [{
  "MemBeginTime": 1,
  "MemEndTime": 2,
  "PronAccuracy": 65,
  "PronFluency": 0.3,
  "Word": "xxx",
  "MatchTag": 1,
  "PhoneInfos": [{
    "MemBeginTime": 1,
    "MemEndTime": 2,
    "PronAccuracy": 52,
    "Phone": "b",
    "Stress": true,
    "DetectedStress": false
  }]
}]
}
```

## 5. 开发者资源

### API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

### SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

### 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
-----	----

错误码	描述
InternalServerError	内部错误
InternalServerError.AuthorizeError	服务未开通或已欠费。
InternalServerError.BASE64DecodeFailed	BASE64解码错误
InternalServerError.CannotFindSession	评估之前没有初始化或已过期。
InternalServerError.FailToDecodeVoice	语音解码失败
InternalServerError.IllegalServerResponse	服务器应答非法
InternalServerError.InitialParameterError	初始化参数错误
InternalServerError.InvalidSeqId	分片序号错误。
InternalServerError.InvalidWAVHeader	WAV头部格式非法或不在同一分片内。
InternalServerError.NoConversationFound	使用的会话没有找到或已经被释放
InternalServerError.NoInitBeforeEvaluation	评估之前没有初始化
InternalServerError.ServerInternalError	服务器内部错误
InternalServerError.ServerOverload	服务器过载
InternalServerError.ServiceTimeout	服务超时
InternalServerError.ShardNoStartWithOne	分片序号错误应该从1开始
InternalServerError.StreamingvoicepkgTimeout	流式语音包超时
InternalServerError.VoiceMsgOversized	语音数据大于1MB
InternalServerError.WordLengthTooLong	文本单词超过限制
InvalidParameter	参数错误
InvalidParameter.AuthorizeError	服务未开通或已欠费。
InvalidParameterValue	参数取值错误
InvalidParameterValue.BASEDecodeFailed	BASE64解码错误。
InvalidParameterValue.InvalidSeqId	分片序号错误。
InvalidParameterValue.InvalidWAVHeader	WAV头部格式非法或不在同一分片内。
InvalidParameterValue.NoDocInList	表单中没有文件。
InvalidParameterValue.RefTxtTooLang	输入文本太长。
InvalidParameterValue.ShardNoStartWithOne	分片序号错误应该从1开始。
InvalidParameterValue.StreamingvoicepkgTimeout	流式语音包超时。

错误码	描述
InvalidParameterValue.VadNotDetectedSpeak	没有检测到语音。
InvalidParameterValue.WordLengthTooLong	文本单词超过限制。
ResourceUnavailable.AuthorizeError	服务未开通或已欠费。
ResourceUnavailable.CannotFindSession	评估之前没有初始化或已过期。
ResourceUnavailable.InitStreamNotSupport	该接口不支持init_stream。
ResourceUnavailable.InitStreamUnfinished	初始化请求未完成，请稍后重试。
ResourceUnavailable.LastSeqUnfinished	前一个分片未处理完，请稍后重试。
ResourceUnavailable.NoConversationFound	使用的会话没有找到或已经被释放。
ResourceUnavailable.NoInitBeforeEvaluation	评估之前没有初始化。

# 发音数据传输接口附带初始化过程

最近更新时间：2019-01-18 15:58:33

## 1. 接口描述

接口请求域名：soe.tencentcloudapi.com。

初始化并传输音频数据，分片传输时，尽量保证SeqId顺序传输。音频源目前仅支持16k采样率16bit单声道编码方式，如有不一致可能导致评估不准确或失败。

默认接口请求频率限制：10000次/秒。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：TransmitOralProcessWithInit
Version	是	String	公共参数，本接口取值：2018-07-24
Region	否	String	公共参数，本接口不需要传递此参数。
SeqId	是	Integer	流式数据包的序号，从1开始，当IsEnd字段为1后后续序号无意义，当IsLongLifeSession不为1且为非流式模式时无意义。
IsEnd	是	Integer	是否传输完毕标志，若为0表示未完毕，若为1则传输完毕开始评估，非流式模式下无意义。
VoiceFileType	是	Integer	语音文件类型 1:raw, 2:wav, 3:mp3(三种格式目前仅支持16k采样率16bit编码单声道，如有不一致可能导致评估不准确或失败)。
VoiceEncodeType	是	Integer	语音编码类型 1:pcm。
UserVoiceData	是	String	当前数据包数据，流式模式下数据包大小可以按需设置，数据包大小必须 $\geq 4K$ ，且必须保证分片帧完整（16bit的数据必须保证音频长度为偶数），编码格式要求为BASE64。
SessionId	是	String	语音段唯一标识，一个完整语音一个SessionId。
RefText	是	String	被评估语音对应的文本，句子模式下不超过个 20 单词或者中文文字，段落模式不超过 120 单词或者中文文字，中文评估使用 utf-8 编码，自由说模式该值传空。
WorkMode	是	Integer	语音输入模式，0：流式分片，1：非流式一次性评估
EvalMode	是	Integer	评估模式，0：词模式，1：句子模式，2：段落模式，3：自由说模式，当为词模式评估时，能够提供每个音节的评估信息，当为句子模式时，能够提供完整度和流利度信息。

参数名称	必选	类型	描述
ScoreCoeff	是	Float	评价苛刻指数，取值为[1.0 - 4.0]范围内的浮点数，用于平滑不同年龄段的分数，1.0为小年龄段，4.0为最高年龄段
SoeAppld	否	String	业务应用ID，与账号应用APPID无关，是用来方便客户管理服务的参数，新的SoeAppld 可以在 <a href="#">控制台</a> 【应用管理】下新建。
StorageMode	否	Integer	音频存储模式，0：不存储，1：存储到公共对象存储，输出结果为该会话最后一个分片TransmitOralProcess 返回结果 AudioUrl 字段。
SentenceInfoEnabled	否	Integer	输出断句中间结果标识，0：不输出，1：输出，通过设置该参数，可以在评估过程中的分片传输请求中，返回已经评估断句的中间结果，中间结果可用于客户端 UI 更新，输出结果为TransmitOralProcess请求返回结果 SentenceInfoSet 字段。
ServerType	否	Integer	评估语言，0：英文，1：中文。
IsAsync	否	Integer	异步模式标识，0：同步模式，1：异步模式。
IsQuery	否	Integer	查询标识，当该参数为1时，该请求为查询请求，请求返回该 Session 评估结果。

### 3. 输出参数

参数名称	类型	描述
PronAccuracy	Float	发音精准度，取值范围[-1, 100]，当取-1时指完全不匹配，当为句子模式时，是所有已识别单词准确度的加权平均值。当为流式模式且请求中IsEnd未置1时，取值无意义
PronFluency	Float	发音流利度，取值范围[0, 1]，当为词模式时，取值无意义；当为流式模式且请求中IsEnd未置1时，取值无意义
PronCompletion	Float	发音完整度，取值范围[0, 1]，当为词模式时，取值无意义；当为流式模式且请求中IsEnd未置1时，取值无意义
Words	Array of <a href="#">WordRsp</a>	详细发音评估结果
SessionId	String	语音段唯一标识，一段语音一个SessionId
AudioUrl	String	保存语音音频文件下载地址
SentenceInfoSet	Array of <a href="#">SentenceInfo</a>	断句中间结果，中间结果是局部最优而非全局最优的结果，所以中间结果有可能和最终整体结果对应部分不一致；中间结果的输出便于客户端UI更新；待用户发音完全结束后，系统会给出一个综合所有句子的整体结果。
Status	String	评估 session 状态，"Evaluating"：评估中、"Failed"：评估失败、"Finished"：评估完成
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

### 4. 示例

#### 示例1 传输语音进行发音评估附带初始化过程



传输语音进行发音评估附带初始化过程

#### 输入示例

```
https://soe.tencentcloudapi.com/?Action=TransmitOralProcessWithInit
&SessionId=stress_test_956938
&VoiceFileType=1
&SeqId=0
&VoiceEncodeType=1
&IsEnd=0
&UserVoiceData=VWtsR1JxeUpBd0JYUVZaRlptMTBJQkFBQUFBQkFBRUFnRDRBQUFCOUFBQUNBQkFBVEVsVFZCb0FBQUJ
KVGtaUFNWTkdWQTRBQUFCTVIYWm1OVFI1TVRrdU1UQXdBR1JoZEdGbWIRTUF5
&WorkMode=0
&RefText=again
&EvalMode=0
&ScoreCoeff=3.5
&<公共请求参数>
```

#### 输出示例

```
{
  "Response": {
    "PronAccuracy": 0.99,
    "PronFluency": 0.99,
    "PronCompletion": 0.99,
    "RequestId": "xxxxxxx",
    "Words": [{
      "MemBeginTime": 1,
      "MemEndTime": 2,
      "PronAccuracy": 65,
      "PronFluency": 0.3,
      "Word": "xxx",
      "MatchTag": 1,
      "PhoneInfos": [{
        "MemBeginTime": 1,
        "MemEndTime": 2,
        "PronAccuracy": 52,
        "Phone": "b",
        "Stress": true,
        "DetectedStress": false
      }]
    }]
  }
}
```

## 5. 开发者资源

### API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

## SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

## 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
InternalError	内部错误
InternalError.BASE64DecodeFailed	BASE64解码错误
InternalError.CannotFindSession	评估之前没有初始化或已过期。
InternalError.FailToDecodeVoice	语音解码失败
InternalError.IllegalServerResponse	服务器应答非法
InternalError.InitStreamNotSupport	该接口不支持init_stream
InternalError.InitStreamUnfinished	初始化请求未完成，请稍后重试。
InternalError.InitialParameterError	初始化参数错误
InternalError.InvalidSeqId	分片序号错误。
InternalError.InvalidWAVHeader	WAV头部格式非法或不在同一分片内。
InternalError.LastSeqUnfinished	前一个分片未处理完，请稍后重试。
InternalError.MP3DecodeFailed	MP3转码发生错误
InternalError.NeedToInit	进行评估之前没有进行初始化
InternalError.NoConversationFound	使用的会话没有找到或已经被释放
InternalError.NoDocInList	表单中没有文件
InternalError.NoError	没有错误

错误码	描述
InternalError.NoInitBeforeEvaluation	评估之前没有初始化
InternalError.RefTxtEmpty	检测到不支持的字符在输入文本
InternalError.RefTxtOov	检测到不支持的字符在输入文本
InternalError.RefTxtTooLang	输入文本太长
InternalError.ServerInternalError	服务器内部错误
InternalError.ServerOverload	服务器过载
InternalError.ServiceTimeout	服务超时
InternalError.ShardNoStartWithOne	分片序号错误应该从1开始
InternalError.StreamProcessFail	流式模式数据包处理过程中间失败
InternalError.StreamProcessTimeOut	流式模式数据包处理超时
InternalError.StreamingvoicepkgTimeout	流式语音包超时
InternalError.TimeOut	获得结果超时
InternalError.TooLongPackage	语音数据包长度超过 1MB
InternalError.VadNotDetectedSpeak	没有检测到语音
InternalError.VoiceMsgOversized	语音数据大于1MB
InternalError.VoiceMsgTooShort	语音时长太短
InternalError.WordLengthTooLong	文本单词超过限制
InvalidParameter	参数错误
InvalidParameter.AuthorizeError	服务未开通或已欠费。
InvalidParameter.InitialParameterError	初始化参数错误。
InvalidParameter.VoiceMsgOversized	语音数据大于1MB。
InvalidParameter.WithoutRealName	用户未实名制认证。
InvalidParameterValue	参数取值错误
InvalidParameterValue.BASEDecodeFailed	BASE64解码错误。
InvalidParameterValue.InvalidSeqId	分片序号错误。
InvalidParameterValue.InvalidWAVHeader	WAV头部格式非法或不在同一分片内。
InvalidParameterValue.NoDocInList	表单中没有文件。
InvalidParameterValue.RefTxtTooLang	输入文本太长。

错误码	描述
InvalidParameterValue.ShardNoStartWithOne	分片序号错误应该从1开始。
InvalidParameterValue.StreamingvoicepkgTimeout	流式语音包超时。
InvalidParameterValue.VadNotDetectedSpeak	没有检测到语音。
InvalidParameterValue.WordLengthTooLong	文本单词超过限制。
ResourceNotFound.AppidNotFount	appid不存在。
ResourceNotFound.InterfaceNotFount	接口不存在。
ResourceUnavailable.AuthorizeError	服务未开通或已欠费。
ResourceUnavailable.CannotFindSession	评估之前没有初始化或已过期。
ResourceUnavailable.InitStreamUnfinished	初始化请求未完成，请稍后重试。
ResourceUnavailable.LastSeqUnfinished	前一个分片未处理完，请稍后重试。
ResourceUnavailable.NoConversationFound	使用的会话没有找到或已经被释放。
ResourceUnavailable.NoInitBeforeEvaluation	评估之前没有初始化。

## 数据结构

最近更新时间：2019-01-18 15:58:34

### PhoneInfo

单音节评价结果

被如下接口引用：TransmitOralProcess、TransmitOralProcessWithInit。

名称	类型	描述
MemBeginTime	Integer	当前音节语音起始时间点，单位为ms
MemEndTime	Integer	当前音节语音终止时间点，单位为ms
PronAccuracy	Float	音节发音准确度，取值范围[-1, 100]，当取-1时指完全不匹配
DetectedStress	Boolean	当前音节是否检测为重音
Phone	String	当前音节
Stress	Boolean	当前音节是否应为重音

### SentenceInfo

语音过程中断句的中间结果

被如下接口引用：TransmitOralProcess、TransmitOralProcessWithInit。

名称	类型	描述
SentenceId	Integer	句子序号，在段落、自由说模式下有效，表示断句序号，最后的综合结果的为-1。
Words	Array of <a href="#">WordRsp</a>	详细发音评估结果
PronAccuracy	Float	发音精准度，取值范围[-1, 100]，当取-1时指完全不匹配，当为句子模式时，是所有已识别单词准确度的加权平均值。当为流式模式且请求中IsEnd未置1时，取值无意义
PronFluency	Float	发音流利度，取值范围[0, 1]，当为词模式时，取值无意义；当为流式模式且请求中IsEnd未置1时，取值无意义
PronCompletion	Float	发音完整度，取值范围[0, 1]，当为词模式时，取值无意义；当为流式模式且请求中IsEnd未置1时，取值无意义

### WordRsp

单词评分细则

被如下接口引用：TransmitOralProcess、TransmitOralProcessWithInit。

名称	类型	描述
MemBeginTime	Integer	当前单词语音起始时间点，单位为ms
MemEndTime	Integer	当前单词语音终止时间点，单位为ms
PronAccuracy	Float	单词发音准确度，取值范围[-1, 100]，当取-1时指完全不匹配
PronFluency	Float	单词发音流利度，取值范围[0, 1]
Word	String	当前词
MatchTag	Integer	当前词与输入语句的匹配情况，0:匹配单词、1：新增单词、2：缺少单词
PhoneInfos	Array of <a href="#">PhoneInfo</a>	音节评估详情

# 错误码

最近更新时间：2019-01-18 15:58:34

## 功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

## 错误码列表

### 公共错误码

错误码	说明
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。请在控制台检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格。
AuthFailure.SignatureExpire	签名过期。Timestamp 和服务器时间相差不得超过五分钟，请检查本地时间是否和标准时间同步。
AuthFailure.SignatureFailure	签名错误。签名计算错误，请对照调用方式中的接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。

错误码	说明
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s)请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

## 业务错误码

错误码	说明
InternalError	内部错误
InternalError.AuthorizeError	服务未开通或已欠费。
InternalError.BASE64DecodeFailed	BASE64解码错误
InternalError.CannotFindSession	评估之前没有初始化或已过期。
InternalError.FailToDecodeVoice	语音解码失败
InternalError.IllegalServerResponse	服务器应答非法
InternalError.InitStreamNotSupport	该接口不支持init_stream
InternalError.InitStreamUnfinished	初始化请求未完成，请稍后重试。
InternalError.InitialParameterError	初始化参数错误
InternalError.InvalidSeqId	分片序号错误。
InternalError.InvalidWAVHeader	WAV头部格式非法或不在同一分片内。
InternalError.LastSeqUnfinished	前一个分片未处理完，请稍后重试。



错误码	说明
InternalError.MP3DecodeFailed	MP3转码发生错误
InternalError.NeedToInit	进行评估之前没有进行初始化
InternalError.NoConversationFound	使用的会话没有找到或已经被释放
InternalError.NoDocInList	表单中没有文件
InternalError.NoError	没有错误
InternalError.NoInitBeforeEvaluation	评估之前没有初始化
InternalError.RefTxtEmpty	检测到不支持的字符在输入文本
InternalError.RefTxtOov	检测到不支持的字符在输入文本
InternalError.RefTxtTooLang	输入文本太长
InternalError.ServerInternalError	服务器内部错误
InternalError.ServerOverload	服务器过载
InternalError.ServiceTimeout	服务超时
InternalError.ShardNoStartWithOne	分片序号错误应该从1开始
InternalError.StreamProcessFail	流式模式数据包处理过程中间失败
InternalError.StreamProcessTimeOut	流式模式数据包处理超时
InternalError.StreamingvoicepkgTimeout	流式语音包超时
InternalError.TimeOut	获得结果超时
InternalError.TooLongPackage	语音数据包长度超过 1MB
InternalError.VadNotDetectedSpeak	没有检测到语音
InternalError.VoiceMsgOversized	语音数据大于1MB
InternalError.VoiceMsgTooShort	语音时长太短
InternalError.WordLengthTooLong	文本单词超过限制
InvalidParameter	参数错误
InvalidParameter.AuthorizeError	服务未开通或已欠费。
InvalidParameter.InitialParameterError	初始化参数错误。
InvalidParameter.VoiceMsgOversized	语音数据大于1MB。
InvalidParameter.WithoutRealName	用户未实名制认证。
InvalidParameterValue	参数取值错误

错误码	说明
InvalidParameterValue.BASEDecodeFailed	BASE64解码错误。
InvalidParameterValue.InvalidSeqId	分片序号错误。
InvalidParameterValue.InvalidWAVHeader	WAV头部格式非法或不在同一分片内。
InvalidParameterValue.NoDocInList	表单中没有文件。
InvalidParameterValue.RefTxtTooLang	输入文本太长。
InvalidParameterValue.ShardNoStartWithOne	分片序号错误应该从1开始。
InvalidParameterValue.StreamingvoicepkgTimeout	流式语音包超时。
InvalidParameterValue.VadNotDetectedSpeak	没有检测到语音。
InvalidParameterValue.WordLengthTooLong	文本单词超过限制。
ResourceNotFound.AppidNotFount	appid不存在。
ResourceNotFound.InterfaceNotFount	接口不存在。
ResourceUnavailable.AuthorizeError	服务未开通或已欠费。
ResourceUnavailable.CannotFindSession	评估之前没有初始化或已过期。
ResourceUnavailable.InitStreamNotSupport	该接口不支持init_stream。
ResourceUnavailable.InitStreamUnfinished	初始化请求未完成，请稍后重试。
ResourceUnavailable.LastSeqUnfinished	前一个分片未处理完，请稍后重试。
ResourceUnavailable.NoConversationFound	使用的会话没有找到或已经被释放。
ResourceUnavailable.NoInitBeforeEvaluation	评估之前没有初始化。